

CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A method comprising:
receiving, by an application executed by an operating system, a plurality of operating parameters having values describing a plurality of different types of resources of a client device;
determining a value representing ~~a performance measure~~ an estimate of current load on the client device based at least in part on a combination of the plurality of operating parameter values describing the plurality of different types of resources of the client device;
assigning the value representing the ~~performance measure~~ estimate of current load on the client device to a usage variable, ~~wherein the usage variable value defines a current usage of a particular combination of resources of the client device;~~ and
correlating by the application a ~~resource usage level~~ of throttling of the application with the usage variable, the correlating comprising:
examining a representation of a mapping of ~~usage variable values~~ estimates of load on the client device to ~~resource usage~~ levels of throttling, wherein each tuple in the mapping ~~specifies~~ maps a particular ~~value of the usage variable and~~ estimate of load on the client device to a particular ~~resource usage level~~ of throttling;

identifying a tuple of the mapping for which the particular value of the estimate of load on the client device usage variable matches the estimate of current load on the client device value assigned to the usage variable; and the application modifying its own execution to throttle its usage to the use the particular resource usage level of throttling specified by the identified tuple.

2. (Previously Presented) The method of claim 1, wherein the application modifying its own execution comprises the application suspending one or more operations when the value assigned to the usage variable exceeds a threshold.

3. (Previously Presented) The method of claim 1, wherein the application modifying its own execution comprises the application performing an activity affecting the usage variable within a threshold time of the usage variable indicating that the client device is performing an existing activity.

4. (Previously Presented) The method of claim 1, wherein the application modifying its own execution comprises the application adjusting a rate of operation based at least in part on the value assigned to the usage variable.

5. (Previously Presented) The method of claim 1, wherein the application modifying its own execution comprises the application adjusting a sequence of operations based at least in part on the value assigned to the usage variable.

6. (Previously Presented) The method of claim 1, wherein the application modifying its own execution comprises the application adjusting an active feature based at least in part on the value assigned to the usage variable.

7. (Previously Presented) The method of claim 1, wherein the client device comprises a client processor and a client memory storage device.

8. (Previously Presented) The method of claim 1, wherein receiving the plurality of operating parameters comprises monitoring at least one of the plurality of the operating parameters.

9. (Previously Presented) The method of claim 1, further comprising monitoring a period of inactivity of the client device.

10. (Previously Presented) The method of claim 7, wherein receiving the plurality of operating parameters comprises receiving at least one of the plurality of operating parameters during an initial load of the client processor.

11. (Previously Presented) The method of claim 1, wherein receiving the plurality of operating parameters comprises receiving at least one of the plurality of operating parameters during a predetermined time interval.

12. (Previously Presented) The method of claim 11, wherein at least one of the plurality of operating parameters comprises a client processor load.

13-14. (Canceled)

15. (Currently Amended) The method of claim 7, further comprising writing to a non-transitory computer readable storage medium of the client memory storage device.

16. (Previously Presented) The method of claim 7, wherein the plurality of operating parameters comprise a first parameter and a second parameter, the first parameter comprising a speed of the client processor and the second parameter comprising a capacity of the client memory storage device.

17. (Currently Amended) A non-transitory computer readable storage medium comprising instructions, that, when executed, cause an application to perform steps comprising:

- receiving, by an application executed by an operating system, a plurality of operating parameters having values describing a plurality of different types of resources of a client device;
- determining a value representing a ~~performance measure~~ an estimate of current load on the client device based at least in part on a combination of the plurality of operating parameter values describing the plurality of different types of resources of the client device;
- assigning the value representing the ~~performance measure~~ estimate of current load on the client device to a usage variable, ~~wherein the usage variable value defines a current usage of a particular combination of resources of the client device;~~
- and
- correlating by the application a ~~resource usage level~~ of throttling of the application with the usage variable, the correlating comprising:

examining a representation of a mapping of ~~usage variable values~~ estimates of load on the client device to ~~resource usage~~ levels of throttling, wherein each tuple in the mapping ~~specifies~~ maps a particular value of the ~~usage variable and~~ estimate of load on the client device to a particular ~~resource usage~~ level of throttling;

identifying a tuple of the mapping for which the particular value of the estimate of load on the client device ~~usage variable~~ matches the estimate of current load on the client device ~~value assigned to the usage variable~~; and

the application modifying its own execution to throttle its usage to the ~~use the~~ ~~particular resource usage~~ level of throttling specified by the identified tuple.

18. (Currently Amended) The non-transitory computer readable storage medium of claim 17, wherein the application modifying its own execution comprises the application suspending one or more operations when the value assigned to the usage variable exceeds a threshold.

19. (Currently Amended) The non-transitory computer readable storage medium of claim 17, wherein the application modifying its own execution comprises the application performing an activity affecting the usage variable within a threshold time of the usage variable indicating that the client device is performing an existing activity.

20. (Currently Amended) The non-transitory computer readable storage medium of claim 17, wherein the application modifying its own execution comprises the application adjusting a rate of operation based at least in part on the value assigned to the usage variable.

21. (Currently Amended) The non-transitory computer readable storage medium of claim 17, wherein the application modifying its own execution comprises the application adjusting a sequence of operations based at least in part on the value assigned to the usage variable.

22. (Currently Amended) The non-transitory computer readable storage medium of claim 17, wherein the application modifying its own execution comprises the application adjusting an active feature based at least in part on the value assigned to the usage variable.

23. (Currently Amended) The non-transitory computer readable storage medium of claim 17, wherein the client device comprises a client processor and a client memory storage device.

24. (Currently Amended) The non-transitory computer readable storage medium of claim 17, further comprising instructions, that, when executed, cause the application to perform the step of monitoring a period of inactivity of the client device.

25. (Currently Amended) The non-transitory computer readable storage medium of claim 23, wherein receiving the plurality of operating parameters comprises receiving at least one of the plurality of operating parameters during an initial load of the client processor.

26. (Currently Amended) The non-transitory computer readable storage medium of claim 23, wherein receiving the plurality of operating parameters comprises receiving at least one of the plurality of operating parameters during a predetermined time interval.

27. (Currently Amended) The non-transitory computer readable storage medium of claim 26, wherein at least one of the plurality of operating parameters comprises a client processor load.

28. (Canceled)

29. (Canceled)

30. (Currently Amended) The non-transitory computer readable storage medium of claim 23, further comprising instructions, that, when executed, cause the application to perform the step of writing to a non-transitory computer readable storage medium of the client memory storage device.

31. (Currently Amended) The non-transitory computer readable storage medium of claim 23, wherein the plurality of operating parameters comprise a first parameter and a second parameter, the first parameter comprising a speed of the client processor and the second parameter comprising a capacity of the client memory storage device.

32. (Currently Amended) The non-transitory computer readable storage medium of claim 17, wherein receiving the plurality of operating parameters comprises monitoring at least one of the plurality of operating parameters.

33. (Previously Presented) The method of claim 1, wherein the usage variable is a quantitative performance measure of the client device.

34. (Previously Presented) The method of claim 1, wherein the usage variable is a qualitative performance measure of the client device.

35. (Previously Presented) The method of claim 1, wherein the application modifying its own execution further comprises the application throttling back its usage of the client device.

36. (Previously Presented) The method of claim 1, wherein the application dynamically modifies its own execution based on dynamic changes to the value assigned to the usage variable.

37. (Previously Presented) The method of claim 1, wherein the application modifying its own execution further comprises the application pausing between execution of resource-intensive calculations.

38. (Previously Presented) The method of claim 1, wherein a resource used by the application is memory and wherein the application modifying its own execution further comprises the application dynamically scaling back its memory usage based on dynamic changes to the value assigned to the usage variable.

39. (Previously Presented) The method of claim 1, wherein a resource used by the application is network bandwidth and wherein the application modifying its own execution further comprises the application throttling-back usage of network bandwidth based on dynamic changes to the value assigned to the usage variable.

40. (Previously Presented) The method of claim 1, wherein the representation of the mapping comprises a table having a column representing the usage variable and a column representing the resource usage level of the application and each tuple in the mapping corresponds to a row in the table.

41. (Previously Presented) The method of claim 1, wherein there are a plurality of usage variables and wherein the correlating comprises the application modifying its own execution based at least in part on changes to values assigned to the plurality of usage variables.

42. (Canceled)

43. (Currently Amended) The non-transitory computer readable storage medium of claim 17, wherein the representation of the mapping comprises a table having a column representing the usage variable and a column representing the resource usage level of the application and each tuple in the mapping corresponds to a row in the table.

44. (Currently Amended) A method comprising:

receiving, by an application executed by an operating system, a plurality of operating parameters having values describing a plurality of different types of resources of a client device;

determining a ~~usage variable~~ value representing a ~~performance measure~~ an estimate of current load on the client device, ~~the usage variable value~~ based at least in part on a combination of the plurality of operating parameter values;

examining a representation of a mapping comprising tuples, each tuple specifying a ~~usage variable value and a resource usage level~~ mapping from an estimate of

load on the client device to a level of throttling, wherein the ~~usage variable~~
~~value defines a current~~ estimate of load specified in a tuple is based on a usage
of a particular combination of resources of the client device;

identifying a tuple, in the representation of the mapping, for which ~~a usage variable~~
value the particular value of the estimate of load on the client device specified
in the tuple matches the value representing the estimate of current load on the
client device ~~value assigned to the usage variable~~; and

modifying, by the application, execution of the application to throttle its usage to the
use the resource usage level of throttling specified by [[of]] the identified
tuple.

45. (Previously Presented) The method of claim 1, wherein the application modifying its own execution comprises the application combining an activity performed by the application with another activity performed on the client device, the combining reducing resources of the client device consumed by the combined activities.

46. (Previously Presented) The method of claim 45, wherein the combined activities are disk activities.